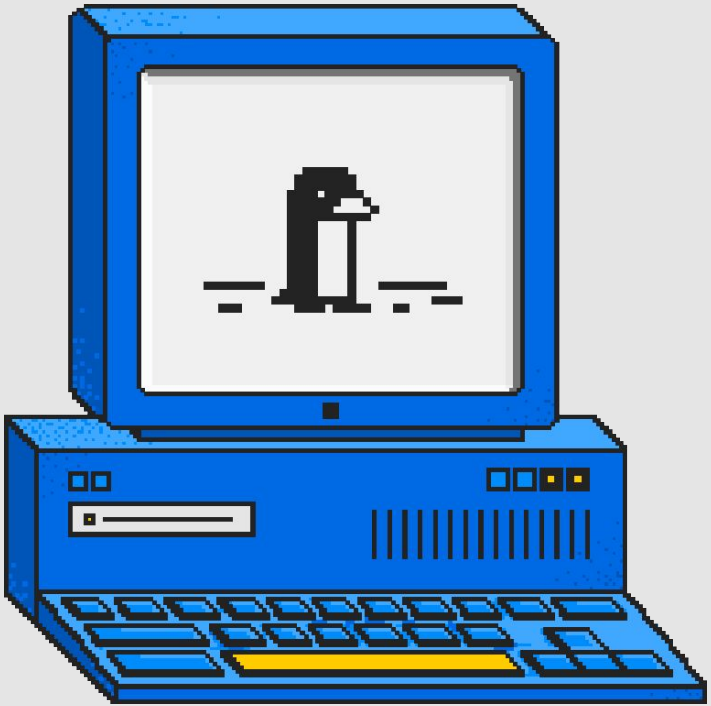


Mamy AI w domu!

Tomasz Ławicki

JA: MAMO KUPIMY  ?

MAMA:  MAMY W DOMU

 W DOMU:

Tomasz Ławicki

- Były Przewodniczący LAG
- Student Systemów rozproszonych i chmurowych
- Nix fan boy
- Pasionat Klawiatur

“Wykorzystanie chatbota z technologią RAG do wspomagania pracy serwisantów w aplikacji iService”

- ❏ RAG
- ❏ Localny LLM
- ❏ Mistral 7B
- ❏ NVIDIA Tesla P100 16GB
- ❏ sentence transformers

LAG

LAG

Agenda

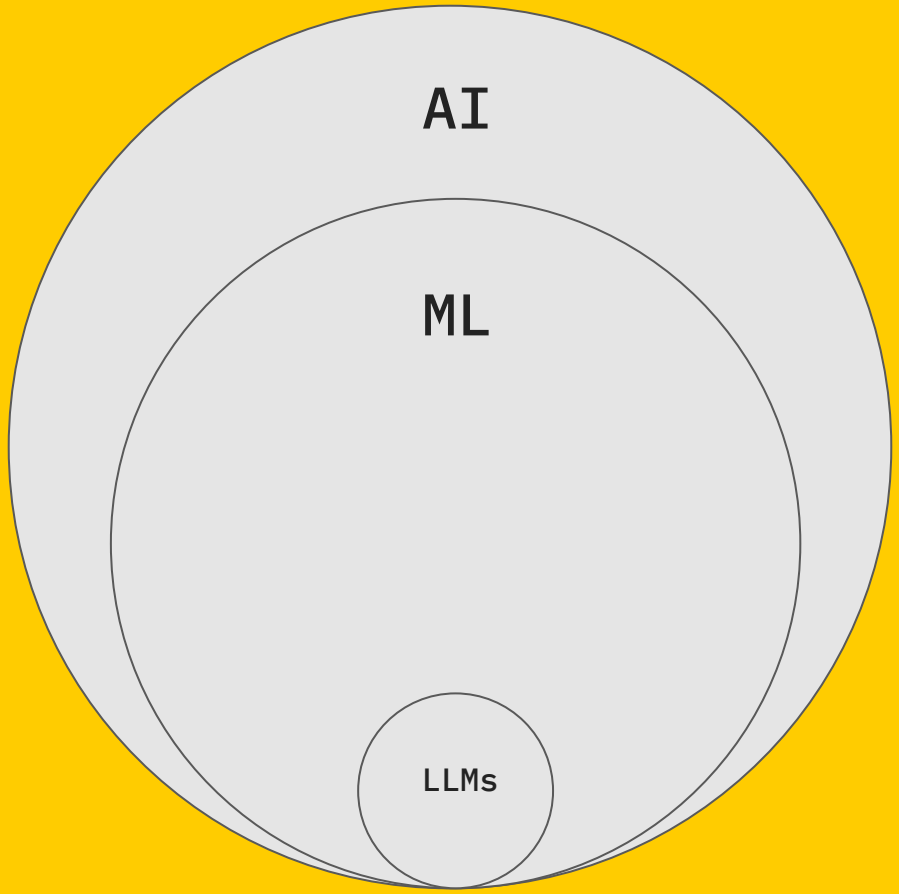
- ❏ Wstęp
- ❏ Modele
- ❏ Inferencja
- ❏ Benchmark
- ❏ Zastosowania

LAG

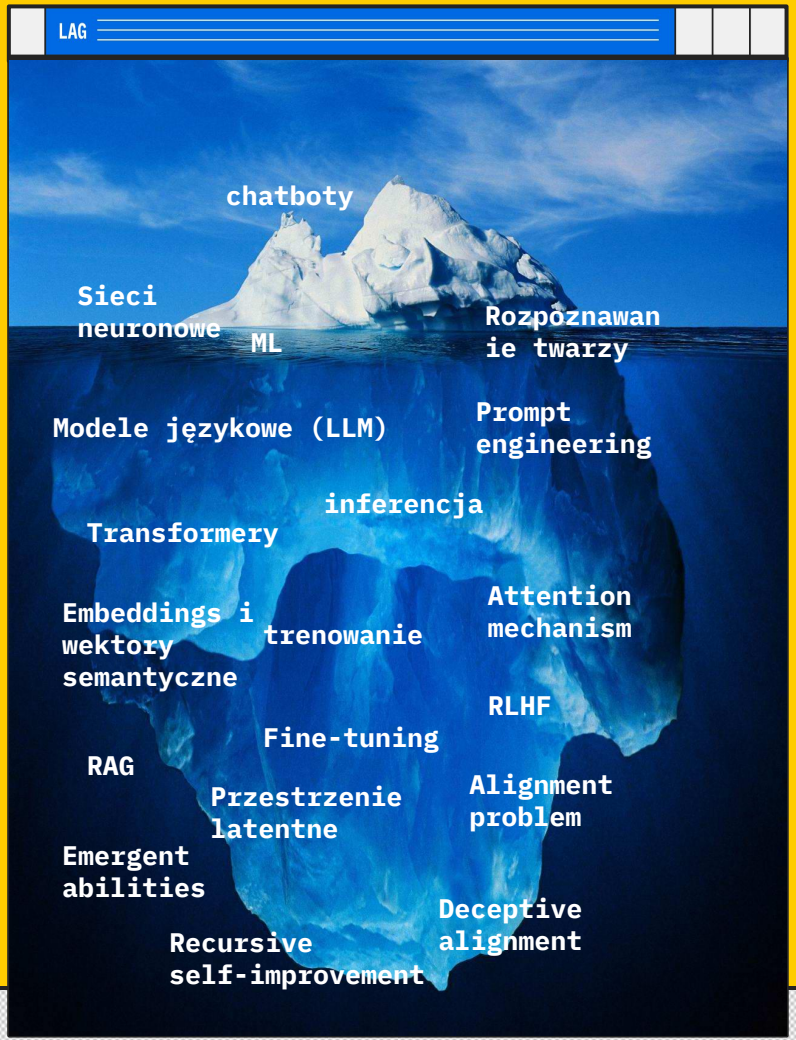
DEMO

ollama + opencode

LLM != AI




LAG
LLM


LAG
chatboty
Sieci neuronowe
ML
Rozpoznawanie twarzy
Modele językowe (LLM)
Prompt engineering
inferencja
Transformery
Attention mechanism
Embeddings i wektory semantyczne
trenowanie
RLHF
Fine-tuning
RAG
Alignment problem
Przestrzenie latentne
Emergent abilities
Deceptive alignment
Recursive self-improvement

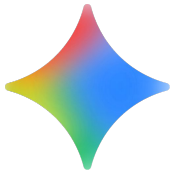
open weight != open source

LAG

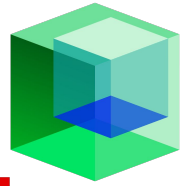
closed source

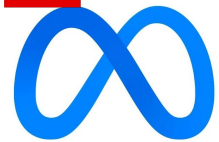
OpenAI

ANTHROP\C



open weight





open source

Hardware

- GPU
- CPU + unified memory
- CPU + RAM

- Rozmiar pamięci:
  GB RAM/VRAM/UMA
- Przepustowość pamięci:
  GB/s lub TB/s
- Moc obliczeniowa:
  TFLOPS/TOPS

AMD RX 7900 XT 20GB ~ 3100 zł

Model	Zalecane	Unified memory vs CPU	GPU VRAM vs CPU
3B–4B Q4	6 - 8 GB	2-5×	3 - 10×
7B–8B Q4	8 - 12 GB	3-6×	5 - 15×
13B–14B Q4	16 - 24 GB	3 - 8×	5 - 20×
20B–27B Q4	24 - 40GB	3 - 8×	5 - 20×
30B–34B Q4	32 - 48 GB	4 - 10×	8 - 25×
70B Q4	64 - 96 GB	5 - 15×	może być 2 - 20× albo gorzej, jeśli spilluje do RAM
100B–200B Q4	96 - 256 GB	UMA wygrywa pojemnością	wygrywa tylko przy multi-GPU/dużym HBM

https://huggingface.co/blog/llama31 https://ollama.com/library https://localllm.in/blog/ollama-vram-requirements-for-local-llms

Jak wybrać swój pierwszy model?

Jak wybrać model?

+

AI models

https://huggingface.co/

B/M	liczba parametrów, np. 7B = 7 miliardów.
Base	model bazowy, zwykle nie najlepszy jako chatbot.
Instruct / IT / Chat	model dostrojony do rozmowy i wykonywania poleceń.
MoE	Mixture of Experts, model aktywujący tylko część „ekspertów” przy generowaniu.
GGUF	popularny format do lokalnego uruchamiania modeli, szczególnie przez llama.cpp, Ollama i LM Studio.
safetensors	popularny format wag w ekosystemie Hugging Face/Transformers.
Q4/Q5/Q6/Q8	poziomy kwantyzacji; niższa liczba oznacza zwykle mniejszy plik i niższe wymagania, ale potencjalnie większą utratę jakości.
Q4_K_M / Q5_K_M	popularne warianty GGUF do lokalnego uruchamiania LLM-ów.
FP16/BF16/FP32	precyzja wag modelu; im wyższa, tym większe zużycie pamięci.
ctx / context length	długość kontekstu, czyli ile tekstu model może brać pod uwagę naraz.
KV cache	dodatkowa pamięć zużywana podczas generowania, szczególnie ważna przy długim kontekście.

llmfit

- Jedna komenda
- Auto detekcja sprzętu
- Baza modeli z ich oceną i szacunkową wydajnością

https://github.com/AlexsJones/llmfit

llm-checker

Zintegrowany z ollama wybór modeli

https://github.com/Pavelevich/llm-checker

Domowa destylacja - Open-R1

Zakładając min 24GB VRAM można z tym zrobić:

- distillation z większego modelu do mniejszego
- generować małe datasety syntetyczne
- benchmarki małych modeli reasoningowych

https://huggingface.co/open-r1

Jak uruchomić swój pierwszy model?

Jak pierwszy model to ollama

- „plug and play”
- potrafi pobrać model
- uruchomić go lokalnie
- wystawić proste API

Na stronie Ollama znajdziemy też bibliotekę dostępnych modeli, razem z wariantami i informacjami potrzebnymi do szybkiego rozpoczęcia pracy.

Jak jak kolejny no to już może ...

LAG

... coś poziom niżej

llama.cpp

- Popularny
- Szczególnie format GGUF
- ograniczona ilość VRAM
- serwer HTTP kompatybilny z API OpenAI

- Intel / Nvidia / AMD GPU
- CPU

https://github.com/ggml-org/llama.cpp

Pobieranie modeli z HF:

```
llama-cli -hf ggml-org/gemma-3-1b-it-GGUF
```

vLLM

LAG

- częściej wybierany do wydajnego serwowania modeli na GPU
- projektowany z myślą o
  - wysokiej przepustowości
  - wielu użytkownikach
  - produkcyjnych zastosowaniach
- serwer HTTP kompatybilny z API OpenAI

- NVIDIA / AMD GPU
- Intel XPU
- Apple Silicon
- CPU

SGLang

LAG

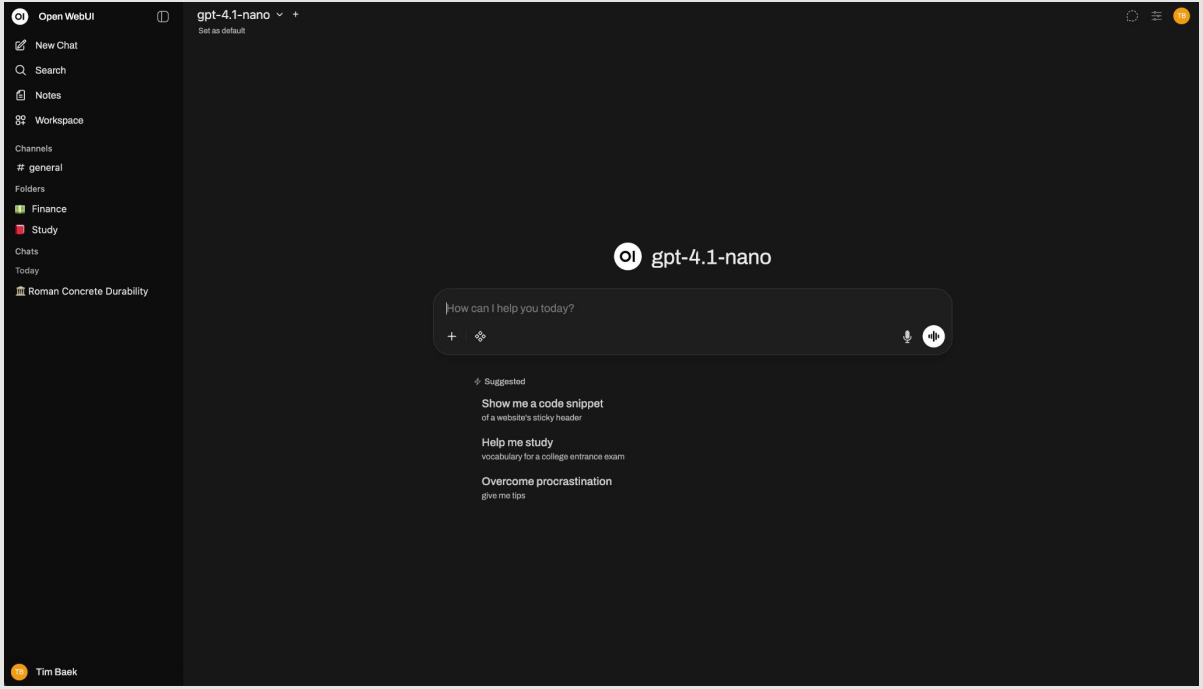
- podobnym kierunku co vLLM
- szybkiego serwowania dużych modeli językowych i multimodalnych
- serwer HTTP kompatybilny z API OpenAI
- AMD GPUs
- Apple Metal
- Intel Xeon CPUs
- Google TPU
- NVIDIA DGX Spark
- NVIDIA Jetson
- Ascend NPUs
- Intel XPU

Jak uruchomić swój pierwszy model? Podsumowanie

Narzędzie	Najlepsze do	Poziom trudności	Typowy scenariusz
Ollama	Szybkiego startu i lokalnego testowania modeli	Niski	„Chcę pobrać model i pogadać z nim lokalnie”
llama.cpp	GGUF, CPU/GPU, maksymalnej kontroli lokalnie	Średni	„Mam model z Hugging Face i chcę sam dobrać parametry”
vLLM	Szybkiego serwowania modeli na GPU	Wyższy	„Chcę wystawić model przez API dla aplikacji lub wielu użytkowników”
SGLang	Wydajnego serwowania LLM/VLM i bardziej złożonych pipeline’ów	Wyższy	„Chcę framework do szybkiego inference i kontroli przepływu”

Open WebUI

- ChatGPT-like
- Możliwość połączenia z poprzednio wymienionymi
- self-hosting



LAG

DEMO

vLLM + Pi

Konkluzja

Co można z tym zrobić?

Zakładając, że chodzi nam Gemma 4 26B-A4B-it uruchomiona lokalnie przez Ollamę

Co można z tym zrobić?

- ❏ Lokalne RAG po własnych notatkach i dokumentach
- ❏ Lokalny asystent do dokumentacji technicznej
- ❏ Lokalny generator testów i checklist
- ❏ Lokalny coding assistant w OpenCode *
- ❏ ...

Czego bym unikał?

- ❏ Dużych zmian kodu
- ❏ Security audytu
- ❏ Samodzielnego agenta
- ❏ Decyzji robionych przez model

Dzielcie pracę na kroki - plan.md

Małe modele najlepiej działają, gdy dostają małe, konkretne i sprawdzalne zadania.

Słodko gorzka prawda

JA: MAMO KUPIMY  ?

MAMA:  MAMY W DOMU

 W DOMU:

Podsumowanie Stacku

Warstwa	Narzędzie	Rola
Model hub	Hugging Face	Znajdź model, model card, licencję i format plików
Dobór do sprzętu	llmfit	Sprawdź, czy model zmieści się w RAM/VRAM
Prosty runtime	Ollama	Pobierz model, uruchom lokalnie, wystaw API
Runtime z kontrolą	llama.cpp	Uruchamiaj GGUF z większą kontrolą parametrów
Aplikacja	OpenCode / Pi / RAG / skrypt	Użyj modelu w realnym workflow

Free-Rat blog

https://free-rat.dev/blog.html

Ollama

https://ollama.com/

Hugging Face

https://huggingface.co/

llmfit

https://github.com/AlexsJones/llmfit

Pi Agent

https://pi.dev/

OpenCode

https://opencode.ai/

DZIĘKUJE ZA UWAGĘ!